REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE April 23, 1999	3. REPORT TYPE AND DATES COVERED Technical Report

4. TITLE AND SUBTITLE

Speech Recognition Using Randomized Relational Decision Trees

5. FUNDING NUMBERS
DAAH04-96-1-0061

6. AUTHOR(S)
Yali Amit
Alejandro Murua

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Chicago
Chicago, IL 60637

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U. S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

ARO 34172.2-MA

11. SUPPLEMENTARY NOTES

The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

12 a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12 b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

ABSTRACT ON PAGE 2

14. SUBJECT TERMS

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OR REPORT	18. SECURITY CLASSIFICATION ON THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Speech Recognition using Randomized Relational Decision Trees

Yali Amit * and Alejandro Murua †

April 23, 1999

Technical Report no. 487
Department of Statistics
University of Chicago

*Department of Statistics, University of Chicago, Chicago, IL, 60637. Supported in part by the Army Research Office under grant DAAH04-96-1-0061 and MURI grant DAAH04-96-1-0445,

†Department of Statistics, University of Washington, Seattle, WA, 98195-4322. Supported in part by the University of Chicago Block Fund.

Abstract

We explore the possibility of recognizing speech signals using a large collection of coarse *acoustic events,* which describe temporal relations between a small number of local features of the spectrogram. The major issue of invariance to changes in duration of speech signal events is addressed by defining temporal relations in a rather coarse manner, allowing for a large degree of slack. The approach is greedy in that it does not offer an "explanation" of the entire signal as the Hidden Markov Models (HMMs) approach does; rather it accesses small amounts of relational information to determine a speech unit or class. This implies that we recognize words as units, without recognizing their subcomponents. Multiple randomized decision trees are used to access the large pool of acoustic events in a systematic manner and are aggregated to produce the classifier.

1 Introduction

Expert human "observers" of bio-acoustic signals, such as speech and bird songs, visualize the information carried in the acoustic waveform in two-dimensional images containing the time-frequency dynamics of the utterances. These images correspond to spectrograms or Log-spectrograms of the signals. The expert human observer (e.g. a phonetician) is somehow able to learn from a collection of spectrograms, acoustic invariants associated to specific units of vocalization (e.g. phonemes in speech, syllables in songs). These acoustic invariants allow the expert to identify the vocalizations present in the signals. Moreover, when learning to identify different vocalizations, human experts seem to focus on the global shape of the spectrograms, i.e. in the *temporal* relations among several local features in both time and frequency. In fact, properties of the gross shape of the spectrum such as the relation among energies at frequency peaks, and the change in energy distribution over time, are postulated to contain acoustic invariants for certain phonetic features of speech [8, p. 188].

In this paper we attempt to address speech recognition from this point of view. In other words, we explore the possibility of recognizing speech signals using a large

collection of coarse *acoustic events,* which describe temporal relations between a small number of local features of the spectrogram. The major issue of invariance to changes in duration of speech signal events is addressed by defining temporal relations in a rather coarse manner, allowing for a large degree of slack. The approach is greedy in that it does not offer an "explanation" of the entire signal as the Hidden Markov Models (HMMs) approach does; rather it accesses small amounts of relational information to determine a speech unit or class. This, of course, implies that we recognize words as units, without recognizing their subcomponents.

This approach connects directly to ideas investigated in previous works on object and shape recognition; see [2, 3]. There, discrimination is obtained through coarse global arrangements of local image tags in the plane. The tags are very stable in the sense that they occur with high probability in certain parts of the shape, even under rather severe deformations. The *spatial* relations among the image tags are defined in a very coarse manner in order to accommodate the required invariance to shape deformations which preserve shape class. These global arrangements provide a very rich family of representations of the gross shape of the different objects, and provide the tools for recursive relational quantization of the space of objects using multiple decision trees.

The basic ingredients are the following. A collection of local tags is defined together with a family of simple pairwise relations between them. For images the local tags are defined in terms of the data in a small neighborhood. An example could be oriented edge information. In acoustic data the local tags are defined in terms of the data in a small time/frequency interval. They are binary variables which detect the presence of a certain frequency in a certain range of energies. The ranges are determined after normalization so some degree of invariance to amplitude is obtained;

they are moderate in size to accommodate for invariance to audio quality. Relations are specified by coarse constraints on locations between the tags. For example in image data the second tag could be constrained to lie in a wedge of some angle with respect to the first. In acoustic data the second tag could be constrained to lie within a certain interval of time relative to the first (e.g. between 100 to 300 milliseconds after the occurrence of the first). Each arrangement is either present in the data or not, and hence defines a binary variable on the data.

An arrangement of tags is a labeled graph: each vertex of the graph corresponds to some tag type, and each edge, to some relation between the two vertices it connects. As labeled graphs, the tag arrangements have a natural partial ordering: each graph precedes any of its direct extensions, involving an additional tag and a relation. Proceeding along one of the paths of this ordering leads to a recursive relational partitioning of the sample space.

Even using a small number of tag types and relations the total number of arrangements (graphs) of say ten tags is huge. Although this rich family of binary variables may contain an enormous amount of information about the class of the data, it can never be computed in its entirety even for a single data point, let alone for a large training set. These variables can only be accessed incrementally in some order. Decision trees offer a very natural way to systematically explore the partial ordering of the arrangements: more complex arrangements are used as splitting rules as the tree grows deeper. The informative features are found at the same time the sample space is being recursively partitioned or quantized. The vast number of arrangements also allows us to use randomization to produce multiple decision trees which are conditionally weakly dependent, and which can be aggregated to produce powerful classifiers [2, 10].

In the context of bird song data [6], we were able to use these ideas to process both segmented and continuous data, and have achieved recognition rates similar to those of HMMs at a massive gain in computational cost, both in training and testing (see Section 6). For spoken digit data we have achieved a recognition rate of 98% on the segmented TI/NIST Digits data.

We have produced comparable classification rates at a gain in computational cost, and perhaps a more transparent classifier in the sense that it is easier to "see what it does". See for example Figures 4 through 7. We do not however directly address the issue of segmentation and continuous speech as HMMs do, and we are still investigating ways to incorporate this classifier to analyze continuous speech. One encouraging point is the fact that the trees based on these loose relational arrangements are very robust to significant error in segmentation (see Section 5). Also it should be emphasized that the specific tags we have chosen may not be the optimal ones for the definition of the acoustic events. Ideas such as those presented in [11], [12] may lead to tags with more information content, and a higher degree of invariance.

This paper is organized as follows. In Section 2, we give a precise definition of the local tags we use; the relations between them are introduced in Section 3. In Section 4, we describe the randomized tree growing procedure. In Section 5, we present a boosting method, and some of our experimental results. Finally, in Section 6, we argue why Decision trees convey a huge gain in computational cost, when compared to HMMs.

2 Local Tags

The acoustic signal can be represented in several forms. A common goal of these representations is to make the time signal more amenable to further processing, thus

entailing some kind of data-reduction and smoothing of the original signal.

Probably the simplest representation of speech is the *spectrogram.* This three-dimensional representation of the acoustic signal describes the frequency dynamics of an utterance over time. It offers a good visualization of the energy content of the frequencies. This is the representation we used in our experiments, but it is not the only possible choice (e.g. a wavelet transformation of the signal, such as the *waveletogram* in [5] could be used as well).

2.1 The spectrogram

The spectrogram can be thought of as a gray-scale image, whose pixel intensities represent the energy content of the frequencies over time.

Our spectrograms are the output of a smoothing procedure over the time-frequency domain. The time axis is divided in consecutive overlapping frames. Within each frame the signal is weighted with a Hamming window; the resulting signal is then Fourier-transformed, giving rise to a vector of frequency energies (the frequency axis). This vector represents an estimate of the spectrum associated to the signal at this particular time frame. The estimated spectrum is further smoothed according to the Bark scale filter-bank of frequencies [15]. The resulting spectrogram can also be thought of as a matrix (X_{tf}), $t = 1, \ldots, T$, $f = 1, \ldots, F$, where t is the time in frames, and f, the frequency bin. T measures the duration or length of the utterances; it varies from utterance to utterance. A simple statistical analysis shows that T approximately follows a Poisson distribution within a specific speech unit; see Figure 1; this observation will be used later in our experiments in Section 5. F is the total number of frequency bins considered; it is kept fixed for all utterances (in our experiments, we set $F = 18$ or 14, according to the sampling rate at which the data were recorded). The variables X_{tf}'s show typical characteristics of log-normality (see Figure 1), and

hence we model them as such. We use overlapping frames of length 25.6 or 32 milliseconds (depending on the sampling rate at which the data were recorded), taken each 5 milliseconds.

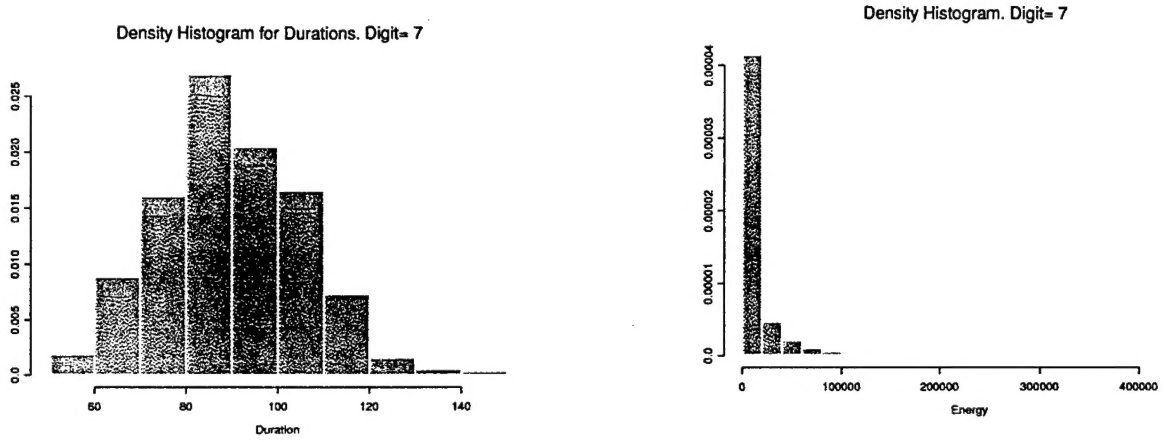

Figure 1: Density histograms for durations and energies of realizations of the digit *seven* in the TI-digits corpus.

2.2 Energy quantization

We believe that the information contained in the spectrogram is very redundant due, in part, to high local correlations, and that the energy content X_{tf} can be quantized in such a way so as to keep the relevant information for recognition intact. Frequency bins with high energy content are important for recognition, since they indicate the presence of frequency peaks (e.g. formants), whose frequency locations and timing within an utterance are believed to be important acoustic features for identification of phonetic features [8, Chapter 8]. It could be argued that high energy content frequency bins do not occur very often in acoustic signals to avoid sending misleading clues to potential decoders (listeners). On the other hand, frequency bins

that are not so crucial for decoding, might experience a fair amount of variation in their energy content, without altering the message. These observations call for a quantization of the energy content that is not uniform, but rather partial towards high energy content, and that at the same time allows for certain degree of slack on moderate values of the energy content. Assuming that $X_{tf} \sim \log \text{Normal}\,(\mu, \sigma^2)$, one can achieve such a quantization scheme by quantizing the values of X_{tf} in $2Q+1$ levels given by distances, in standard deviation units, from the mean μ. Let T_Q denote this quantizing transformation; then we set $T_Q(X_{tf}) = q$ if and only if

$$\frac{1}{2}(q-Q)\sigma \leq \log X_{tf} - \mu < \frac{1}{2}(q-Q+1)\sigma$$

for $q = 0, 1, 2, \ldots, 2Q-1$, and $T_Q(X_{tf}) = 2Q$ if and only if

$$\log X_{tf} - \mu \geq \frac{1}{2}Q\sigma.$$

Notice that the probability between consecutive quantiles associated to this quantization is not constant; in fact

$$p_q = \mathcal{P}(T_Q(X_{tf}) = q) = \Phi(\frac{1}{2}(q-Q+1)) - \Phi(\frac{1}{2}(q-Q)),$$

where $\Phi(\cdot)$ denotes the standard Normal cumulative distribution. For $Q = 4$, the vector of probabilities $\{p_q\}_{q=0}^{9}$ is

$$\{0.044, 0.092, 0.150, 0.191, 0.191, 0.150, 0.092, 0.044, 0.023\}.$$

Also notice that small values of frequency energy are discarded from further processing.

To account for the possible changes in loudness and intensity, we let μ vary with each utterance. However, we assume that σ^2 is the same for all utterances of a particular database. μ could be estimated by the sample average of the logarithm of

the energies of each particular utterance; however, in order to perform the quantization, a normalization of the frequency energies by their sample mean (average) $\overline{X}_{tf}$ is, computationally speaking, more efficient. In fact, since the mean of X_{tf} is $E(X_{tf}) = \exp\{\mu + \sigma^2/2\}$, the normalized variable $X_{tf}/\overline{X}_{tf} \approx X_{tf}/E(X_{tf})$ can be approximated again by a logNormal distribution, but this time the corresponding parameters are $\mu(\text{normalized}) = -\sigma^2/2$, and $\sigma^2(\text{normalized}) = \sigma^2$; hence there is no need to compute the mean of the log energies. The variance σ^2 can be estimated by the pooled (average) sample variance over the whole collection of utterances of the training set.

This quantization scheme is very different from the usual vector quantization technique used with HMMs. In this latter framework, the whole F-dimensional vector space is quantized in about 10^3 regions. In our approach quantization is local in the frequency domain: each component X_{tf} is quantized separately. The number of resulting vector quantization regions is approximately $(2Q+1)^F$, which for moderate Q (e.g. $Q = 4$, $F = 18$), is huge. We never explicitly use multi-frequency quantiles so this never creates a problem. Each of the $2Q+1$ levels at each of the F frequency bins is a "tag", labeled by the quantile and the frequency (f, q). It is a binary feature which is either present or not. Important information regarding co-occurrences of certain frequencies at the same time are represented through the *relations* between the tags, see 3 below.

In cases where the same tag occurs in consecutive time frames we cluster to one tag at the first time of occurrence. The maximal duration of clustering is typically 5 time frames. In the middle panels of Figures 4 through 7 we show the locations of some of the tags, precisely those used in determining certain acoustic events used in a particular tree.

3 Acoustic events

Our procedure is based on the assumption that there are acoustic events that either occur fairly often or rarely on most utterances representing a determined speech unit (class). The presence or absence of several of these acoustic events in a given utterance, gives strong hints for the identification of the speech unit represented by the utterance.

Statistically speaking, the probability of observing the presence or absence of several of these acoustic events is high, given that an utterance is a realization of certain determined speech unit; at the same time, this same probability is fairly small over all utterances (regardless of the speech unit they represent).

3.1 Relations and labeled graphs

We consider particular acoustic events defined by a collection of binary relations between tags in a spectrogram. We note that tags only carry information on energy content at particular frequency bins; hence, tags alone are too primitive features to be relevant for recognition. A tag ℓ_1 is related to another tag ℓ_2 by the time interval I, if the relative time between their occurrences in the spectrogram, $t(\ell_2) - t(\ell_1)$, is contained in the time interval I, i.e. $t(\ell_2) - t(\ell_1) \in I$.

An acoustic event is a *connected* graph. The vertices of the graph are labeled by tag types. The edges between pairs of these vertices are labeled by time intervals defining their relationship. To be precise, let $V = \{\ell_1, \ell_2, \ldots, \ell_k\}$, be a list of tags, and

$$E = \{(\ell_{i,1}, \ell_{i,2}, I_i), i = 1 \ldots, n, \quad \ell_{i,j} \in V,\ j = 1, 2\},$$

be a list of ordered pairs of tags with a time interval. The associated acoustic event

is

$$t(\ell_{i,2}) - t(\ell_{i,1}) \in I_i,\ i = 1, \ldots, n. \tag{1}$$

The value of n denote the *depth* of the event. The condition that the graph is connected implies that there is a path of edges between any two vertices. In other words, for any ℓ_a, ℓ_b in V there are an integer m and a sequence of edges $(\ell_{i_j,1}, \ell_{i_j,2}), j = 1, \ldots, m$, in E such that $\ell_{i_1,1} = \ell_a$, $\ell_{i_m,2} = b$ and $\ell_{i_j,2} = \ell_{i_{j+1},1}$ for $j = 1, \ldots, m-1$

Temporal relations such as the ones given by (1), allow for certain slack in the timing of events on individual utterances; in this way, the sensitivity to the time-warping problem is controlled. We observed in our experiments, that a few non-overlapping intervals suffice in order to obtain good classification rates. We have used the following five intervals (in time frames): $(0, 20), (20, 40), (40, 70), (70, 100), (100, +\infty)$.

Note that on top of the time warping allowed for by the definition of the relations, the acoustic event corresponding to the graph is entirely translation invariant. One can think of particular realizations of these acoustic events as being aligned (warped) to ideal "templates" of the events; these templates correspond to our labeled graphs.

The top panels of Figures 4 through 7 show one instance of such a graph on four different spectrograms of four different digit utterances. The variability in the possible instantiations of this graph is clearly manifest in the four images.

3.2 Information content

One might ask how informative these acoustic events are. For example the distribution on class for those digits containing the acoustic event shown in the top panels of Figures 4 through 7 is given in Table 1 for both the training set (4460) points and the test set (2486) points. One immediately sees that even an event involving 3 tags and 2 relations may be very informative, and an additional 2 tags narrows the distribution

	Prob	Ent	0	1	2	3	4	5	6	7	8	9	10
Ev. 1 (tr)	.08	1.33	.04	.05	0	.04	.14	0	.61	.07	.01	0	0.02
Ev. 1 (te)	.09	1.27	.05	.04	0	.02	.16	.01	.59	.13	0	0	0
Ev. 2 (tr)	.007	.77	0	0	0	.06	.75	0	.15	.03	0	0	0
Ev. 2 (te)	.007	.55	.11	0	0	.05	.83	0	0	0	0	0	0

Table 1: First column: Total probability of event. Second column: Conditional entropy on class given the event (base e). Columns 3-13: Probabilities on class given the presence of the acoustic event. For each event we show the numbers on the training and on test test set. The events correspond to the top and bottom panels of Figure 4, and to nodes 'no,yes,yes' and 'no,yes,yes,yes,yes' in the tree of Figure 2.

effectively to only two classes. The entropy of the prior distribution on class is 2.4 (base e). Note also the strong similarity in the shape of the distributions between test and training even though the conditioning events are rather low probability events. This is an indirect indication that a large degree of invariance is accommodated through the relations and the tag definitions.

3.3 A partial ordering

The number of distinct tags is $(2Q+1)\times F$, and the number of possible binary relations is $N_I \times ((2Q+1)F((2Q+1)F-1))/2$, where N_I is the number of temporal intervals considered. Consequently the total number of acoustic events of depth n with exactly k distinct tags is immense (e.g. if $n = k = 5$, and $N_I = 5$, $Q = 4$, and $F = 18$, on the order of 10^{19}). We therefore need an efficient procedure to explore the pool of acoustic events, with the goal of detecting those relevant for classification. Our approach is constructive, in the sense that events are elucidated vertex by vertex in a suboptimal fashion: an event of depth n is deepened to $n+1$ if the addition of a new vertex and edge, significantly improve the discrimination of the speech units in consideration. Otherwise put, we exploit the partial ordering on the acoustic events inherited from the graphical descriptions. This is done using decision trees as described in the next

section.

4 Decision Trees

The collection of possible arrangements is vast and cannot be precomputed as a binary feature vector. Moreover many arrangements may be useless in terms of the classification problem at hand. Decision trees are used to systematically explore the collection of arrangements and find the most informative ones in terms of classification.

The trees are constructed as follows. At the root of the tree, a search through the collection $\mathcal{G}_0$ of graphs of two vertices is done. Each such graph produces a split in the training data. The graph G_0 yielding the smallest mean conditional entropy on the class variable is chosen. This is the standard splitting criterion for tree growing in the statistics and machine learning literature. Associated to the chosen graph, there is a query, denoted by Q_0, that asks whether a specific acoustic event is present in the spectrogram. Data points for which $Q_0 = 0$ are in the "no" child node. In that node a new search through $\mathcal{G}_0$ is performed in order to find the best split. Data points for which $Q_0 = 1$ are in the "yes" child node, and have one or more instances of the graph G_0, which is now called the pending graph. A search among minimal extensions of this graph is performed to choose the one which leads to the greatest reduction in mean conditional entropy on class. Minimal extensions involve an additional tag in relation to one of the tags in the existing graph.

As the tree is growing, there is a pending graph at each node, determined by all the "yes" answers on the path leading from the root to that node. The only possible splits entertained at the node are minimal extensions of this pending graph. For example the graph of panel 1 in Figure 4 is the pending graph at node 011 (No,Yes,Yes) and also at node 0110 (No,Yes,Yes,No). At node 01101 it is extended to the graph of

panel 5. Tree growing is stopped when not enough data is present at the node to determine a split, or when none of the possible splits yields a significant decrease in entropy. In Figure 2 we show part of a decision tree traversed by the four data points of Figures 4 through 7.

The size of the training set imposes a limit on the depth of a tree and only a very small number of tag arrangements is actually used. More information is accessed by growing randomized multiple trees, where instead of choosing the optimal split among *all* admissible splits (namely minimal extensions), one takes a small random sample of admissible splits and chooses the best among those. This yields somewhat less powerful trees, but the trees are substantially different, and offer complementary "points of view" on the data. Assume N trees are grown $T_1, \ldots, T_N$.

A test data point is dropped down each tree by evaluating the top query Q_0; if the answer is 1 (Yes), the point proceeds to the "yes" node, and the query there is evaluated, and so on until the data point reaches a terminal node. With a slight abuse of notation, we will denote the terminal node reached by a point ω in tree n as $T_n(\omega)$. Consider K classes (speech units). Associated to each terminal node t of a tree, there is a (conditional) probability distribution over the classes, $\mu_t = (\mu_t(1), \ldots, \mu_t(K))$, which has been estimated from the training data. This is called the terminal distribution. When a test point is dropped down N trees, it encounters N such terminal distributions $\mu_{T_1(\omega)}, \ldots, \mu_{T_N(\omega)}$.

The simplest way of aggregating the information in a collection of N trees is to calculate the average distribution

$$\mu(\omega) = \frac{1}{N} \sum_{n=1}^{N} \mu_{T_n(\omega)} = (\mu(\omega, 1), \ldots, \mu(\omega, K)) .$$

The *argmax* of this average distribution is taken to be the classification assigned to

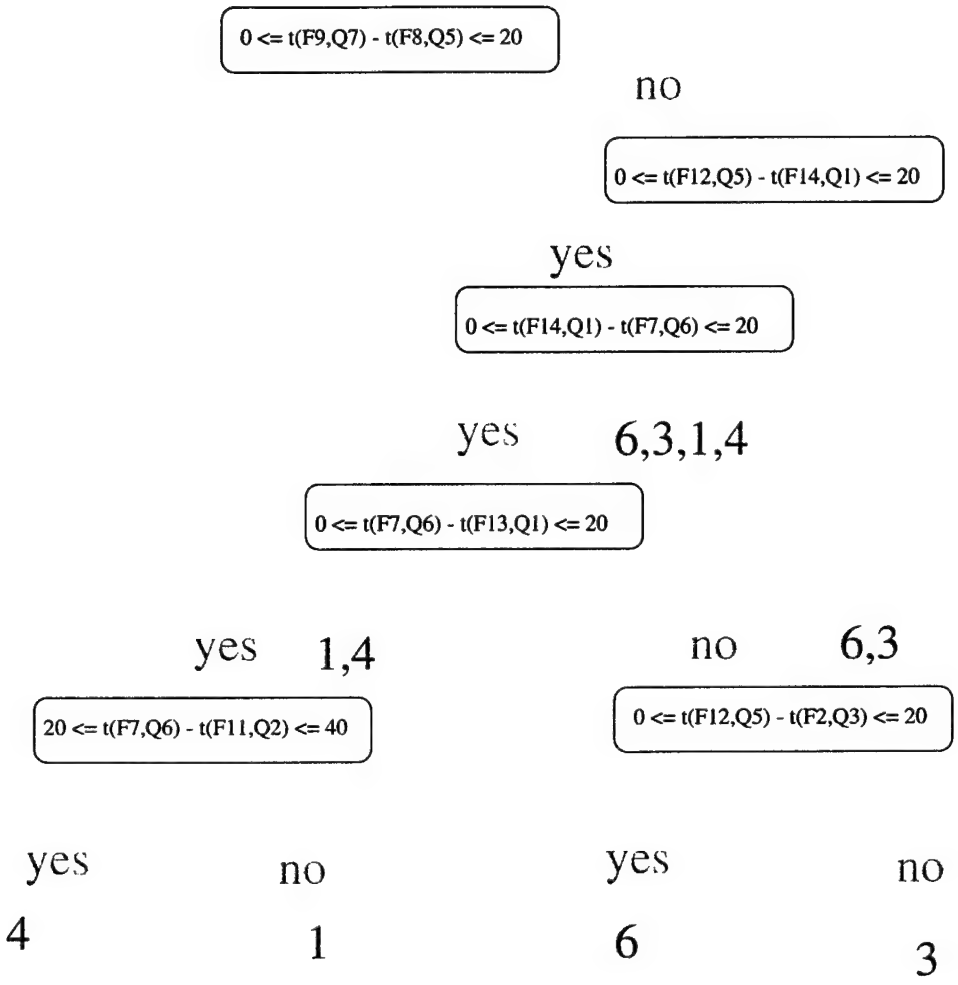

Figure 2: Part of the decision tree traversed by the four data points of figures 4 through 7. The split used at each node is provided explicitly in terms of the tag types and the time interval.

the point ω:

$$C(\omega) = argmax_{k=1,\ldots,K}\{\mu(\omega, k)\}.$$

This classification rule will be referred to as the *Aggregate Distribution Rule* in the following section.

Earlier we indicated that the randomized trees access the data from different points of view. Statistically this translates into the trees being *pairwise weakly dependent* conditional on class. This weak dependence leads to an increase in classification rates when more and more trees are aggregated. A detailed discussion of the properties of this classification rule and of the multiple randomized tree procedure can be found in [2], and [1].

5 Experiments

We applied our procedure to the recognition of isolated or segmented digits (*one, two, three,...,nine,* and *oh*). Our purpose was to find out how well our procedure does over a range of audio quality conditions. Hence, the data consisted of two speech corpora: one containing speech spoken over the telephone, and another one containing studio quality speech.

The telephone data were taken from the CSLU Number Corpus of the Center for Spoken Language Understanding of the Oregon Graduate Institute of Science and Technology. This corpus is a real world application containing "fluent numbers" spoken by thousands of people when saying numbers such as their street address numbers, zip-codes, and telephone numbers. False starts, repetition, and background noise are very common in these data, and make the task difficult (see [4] for details). The corpus is divided into two sets of 8,829, and 6,171 speech files; the first one is reserved for training, and the second one for testing. We located and worked with all

occurrences of the eleven digits in this corpus.

The studio quality speech data were taken from the well-known TI/NIST Connected-Digits Recognition task (also known as TI-digits). This corpus does not include the segmentation of the utterances; hence we hand-segmented a small portion of them; specifically, we hand-segmented those utterances corresponding to digit sequences of at most two digits. Due to this limitation, our training data constituted a very small subset of the data available - 4,460 points in all. Therefore, our results are not directly comparable to previous results reported in the literature. Moreover, our testing data only consisted of 2,486 isolated digits taken from the testing portion of the corpus.

Table 2 shows classification error rates for both corpora, using 100 trees and the aggregate distribution rule (ADR).

Corpus	ADR	ADR+NN
TI-digits	1.89	1.85
Number	8.61	7.93

Table 2: Classification error rates.

In general, recognition improves with the number of trees, but there appears to be a limit to the achievable error rates, as the asymptotes in Figure 3 seem to indicate. Theoretical error rates can be obtained [2, 10] for these type of classifiers; these bounds depend on the average amount of dependency among the decision trees; it is conceivable that in practice due to the limited number of training data, it is impossible to create too many decision trees without building moderate correlation among them.

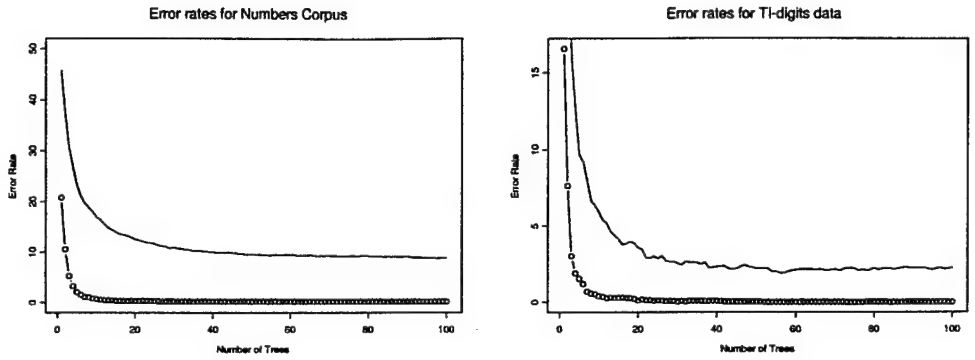

Figure 3: Error rates for increasing number of trees. The solid line corresponds to the testing error rates, and the other line, to the training error rates.

Boosting the classification rates.

We implemented a K-nearest neighbors (K-NN) rule over the space of aggregate distributions resulting from dropping data over the decision trees. The goal was to boost the classification rates with the hope of capturing useful information for recognition from the aggregate average distributions. In fact, we observed that the true digit (class) is among the top two modes of the aggregate average distributions on 96.8% of the data from the Number corpus, and on 99.7% of the data from the TI-digits corpus. This boosting procedure is based on the work in [13]; it consists of selecting a moderate size rejection set from the training data, so as to view the corresponding output aggregate distributions as centroids or prototypes of data that are likely to be rejected, namely data which the aggregate classifier is having trouble classifying. The rejection criterion is based on the ratio between the top two modes of the aggregate average distributions: if the ratio is smaller than certain a priori fixed threshold, then the data point is not recognized, but rejected, instead. About 25% of the training data is rejected in this way. This procedure is also applied to test data, but with a lower threshold, so as to reject at most 10% of the data. Each

rejected data point in the test set is matched to its K-nearest neighbors in the training rejection set, according to the Kullback-Leibler distance between the two aggregate distributions. The data point is then recognized as a realization of the most frequent digit among its K-nearest neighbors. The column named ADR+NN of Table 2 shows the error rates for both corpora after applying the boosting procedure. We obtain an 8% reduction in the error rate for the Number corpus, and about 2% reduction on the error rate for the TI-digits corpus. This indicates that this boosting procedure is more effective on data that are very difficult to discriminate, and hence it might work well on real world speech tasks.

Previous studies [9, 14, 7] with HMM, neural networks, or two-dimensional extensions of HMM, report error classification rates of about 5% to 12% on similar tasks with the Number corpus. Reported error rates for the TI-digits are much smaller than ours (less than 1%); as we mentioned earlier, our results for this corpus are not comparable to those reported in the literature, due to the limitation of our training set.

Cross-Testing.

In order to assess how well our procedure generalizes to data recorded under different quality conditions, we cross-tested the testing portion of the corpora, i.e. the TI-digits were tested with trees trained with the Number corpus, and vice-versa. We observed that correct recognition rates decreased by slightly more than 10% when testing on different audio quality data. But they are still high enough to suggest that somehow our procedure is capturing acoustic events that are invariant across data sets.

Cross-Tested Corpus	ADR
TI-digits	13.50
Number	21.50

Table 3: Cross-Classification error rates.

Sensitivity to segmentation.

To investigate how sensitive our procedure is to erroneous segmentations of word boundaries, we tested our procedure on data whose word (digit) boundaries were randomly marked. In order to randomize the boundaries, we modeled the duration (length) of each digit as a Poisson distribution with certain intensity λ (see Figure 1), depending on the particular digit being considered. The intensities were estimated by the average duration of the digits in the training set. Table 4 shows these estimates for both corpora.

Corpus	0	1	2	3	4	5	6	7	8	9	oh
TI-digits	2.41	1.83	1.62	1.75	1.76	1.92	2.56	2.31	1.52	2.20	1.57
Number	2.58	1.75	1.87	1.92	2.13	2.60	2.68	2.64	1.73	2.25	1.18

Table 4: Poisson Intensities [seconds].

Each utterance ω from the TI-digits testing data set was assigned a duration $d(\omega)$ chosen at random according to the Poisson distribution associated to the corresponding digit. If the random duration $d(\omega)$ was shorter than the actual duration $T(\omega)$ of the utterance, then the utterance was modified by shortening it to $d(\omega)$ time frames. The random segmentation was done by selecting at random from the collection of time frames $\{1, 2, \ldots, T(\omega) - d(\omega)\}$, a starting time (left boundary) for the modified utterance; the ending time (right boundary), was set so that the total duration of the modified utterance was $d(\omega)$. About half the utterances were modified in this manner.

Only modified utterances were tested. This procedure was applied ten times to the available testing data set, yielding the error rates shown in Table 5.

Data	1	2	3	4	5	6	7	8	9	10	Mean	Std. Dev.
Error	1.9	2.1	1.9	2.2	2.5	2.0	2.0	1.9	2.2	2.0	2.0	0.2

Table 5: Error rates for modified boundaries of the TI-digits

The average error rate for the modified boundaries data is almost the same as the error rate of the correctly segmented data; this gives strong evidence that our procedure is not sensitive to small to moderate errors in word segmentation.

6 Computation considerations

It is important to note that our procedure not only produces good classification rates that are comparable to those yielded by HMMs, but also that it requires orders of magnitude fewer calculations (computer operations) both during training and testing.

The following figures were measured on a PC with a Pentium II processor running Cygnus software over Windows NT. We emphasize that the implementation of the algorithm has not been optimized. It took 282 minutes to grow 100 trees of average depth 9.6 with the 4,460 data points comprising the TI-digits corpus (i.e. 2.82 minutes per tree). Testing 2,486 data points from the testing portion of the TI-digits corpus took 14 minutes, i.e. 0.34 seconds per data item. Similarly, it took 752 minutes to grow 100 trees of average depth 12.2 with the 13,264 data points comprising the Number corpus (i.e. 7.52 minutes per tree). Testing 8,642 data points from the testing portion of the Number corpus took 42 minutes, i.e. 0.29 seconds per data point.

Although we do not have the corresponding time measurements for HMMs applied to the same tasks with the same data sets, a comparison between the order of calcu-

lations needed by our procedure and by HMMs to solve these tasks, offer compelling evidence on the computational gain conveyed by our approach.

HMMs. HMMs are notorious for the quantity of CPU time that the training step requires. Let r denote the number of data points in the training set, and d, the average duration of the utterances in the training set. A HMM is characterized by its number of states s, and mixture components m. Each iteration of the Baum-Welch recursion formulas requires $O(s^2md \times r)$ operations (here $O(\cdot)$ stands for the order (magnitude) of the number of operations). On the other hand, ignoring the maximization step to find the best class, testing only requires $O(s^2md \times K)$ operations (recall that K is the number of classes).

Decision Trees. As before, let N denote the total number of trees to be grown. The relevant quantities for a tree are the average depth n of the trees, and the number n_q of randomized queries entertained as admissible splitting rules at each non-terminal node. The selection of an optimal query at each non-terminal node requires $O((K+d) \times n_q)$ operations, since (a) evaluations of the form $t(\ell_1) - t(\ell_2) \in I$, for fixed tags ℓ_1, ℓ_2, over all the locations of these two tags, require $O(d \times \text{length}\,(I))$ operations, and (b) the average entropy of the children nodes requires $O(K)$ operations, since only a fixed number of data points are used in its computation. Since a tree of depth n has at most $2^n - 1$ non-terminal nodes, and at most 2^n leaves, the number of operations needed to grow N trees is of $O(N \times \{2^n(K + d)n_q + 2^n rK\})$. Again, testing only requires $O(ndN + KN)$ operations.

Computational Gain during Testing. ¿From the above calculations, the testing step in both procedures requires about the same number of operations when the

number of trees grown N is about $O(m \times s^2K/(n+K))$. Since usually n and s are rather small numbers, the balancing variable is the number of mixture components m. For large m, decision trees are much faster. In fact, assuming $s = 5$, and $K = 10$, a hundred trees of average depth ten, require about as many operations as ten HMMs with eight mixture components in each state.

Computational Gain during Training. There is no doubt of the enormous gain in computational cost, when using decision trees rather than HMMs. Indeed, from the above calculations, i iterations of the Baum-Welch algorithm for HMMs require $O(K(s^2mdr/K)^i)$ operations, which is a extremely large number of operations, even for $i = 2$, when compared to $O(N2^n(K+d)n_q + N2^nrK)$ operations required for N decision trees. In fact, if we set $s = 5, m = n = 10$, and $n_q = d = 100$, two iterations of the Baum-Welch algorithm require about $O(10r)$ times as many operations as growing 100 trees.

7 Conclusion

Acoustic events based on tags localized in time and frequency, and simple coarse temporal relations, provide informative features for classification of acoustic signals. These events are defined in terms of labeled graphs and inherit a partial ordering. We employ multiple randomized decision trees to access the rich pool of acoustic events, in a systematic way, exploiting the partial ordering to proceed from coarse to fine representations. Time invariance is directly incorporated through the relations, invariance to audio quality is incorporated through the coarse definition of the tags. The learning stage for this approach is much more efficient and transparent than for HMM's. Recognition times are also faster than the more complex HMM models.

On the other hand the issue of segmentation and continuous speech analysis is not addressed.

References

[1] Y. Amit, G. Blanchard, D. Geman, and K. Wilder. Multiple randomized classifiers. Department of Statistics, University of Chicago, 1999.

[2] Y. Amit and D. Geman. Shape quantization and recognition with randomized trees. *Neural Computation*, 9:1545–1588, 1997.

[3] Y. Amit, D. Geman, and K. Wilder. Joint induction of shape features and tree classifiers, *IEEE Trans. PAMI*, 19, 1997.

[4] R. A. Cole, M. Fanty, and T. Lander. Telephone speech corpus development at CSLU. In *Int. Conf. Spoken Language Processing*, 1994.

[5] B. Gidas and A. Murua. Classification and clustering of stop consonants via nonparametric transformations and wavelets. In *Int. Conf. Acoustics, Speech, and Signal Processing*, volume 1, pages 872–875, 1995.

[6] J. Kogan, A. Murua, Y. Amit, D. Margoliash, and R. Larkin. Automated recognition of bird song sound using randomized decision trees. In Preparation.

[7] J. Li and A. Murua. A 2D extended HMM for speech recognition. In *Int. Conf. Acoustics, Speech, and Signal Processing*, 1999.

[8] P. Lieberman and S. Blumstein. Speech Physiology, Speech Perception, and Acoustic Phonetics. Cambridge University Press, 1988.

[9] K. W. Ma. Applying large vocabulary hybrid HMM-MLP methods to telephone recognition of digits and natural numbers. Technical report, International Computer Science Institute, 1995.

[10] A. Murua. Upper bounds for error rates associated to linear combination of classifiers. In Preparation.

[11] P. Niyogi and P. Ramesh. Incorporating voice onset time to improve letter recognition accuracies. In *Int. Conf. Acoustics, Speech, and Signal Processing*, 1998.

[12] P. Niyogi, C. Burges, P. Ramesh. Distinctive feature detection using support vector machines. to appear in *Int. Conf. Acoustics, Speech, and Signal Processing*, 1999.

[13] K. Wilder. *Decision Trees for Shape Recognition.* PhD thesis, University of Massachusets, 1998.

[14] Y. Yan, M. Fanty, and R. Cole. Speech recognition using neural networks with backward-forward probability generated targets. In *Int. Conf. Acoustics, Speech, and Signal Processing*, 1997.

[15] E. Zwicker. Subdivision of the audible frequency range into critical bands (Frequenzgruppen). *J. Acoust. Soc. Amer.*, 33:248, 1961.

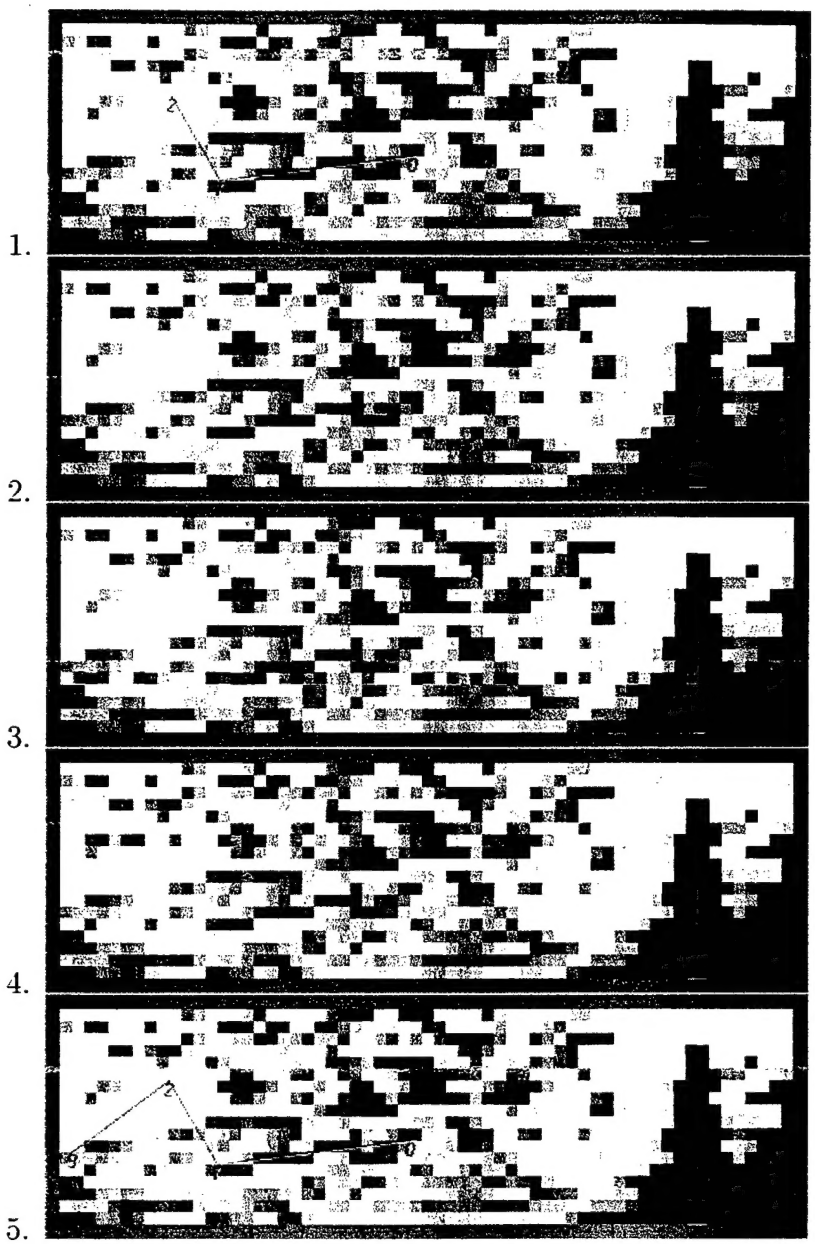

Figure 4: From top to bottom: 1. Graph on a digit '1' at node 011 in a tree. 2. Tags 103 to 107 on the same data. 3. Tags 118 to 123. 4. Tags 58 to 62. 5. Graph on same digit at node 0111 in the same tree.

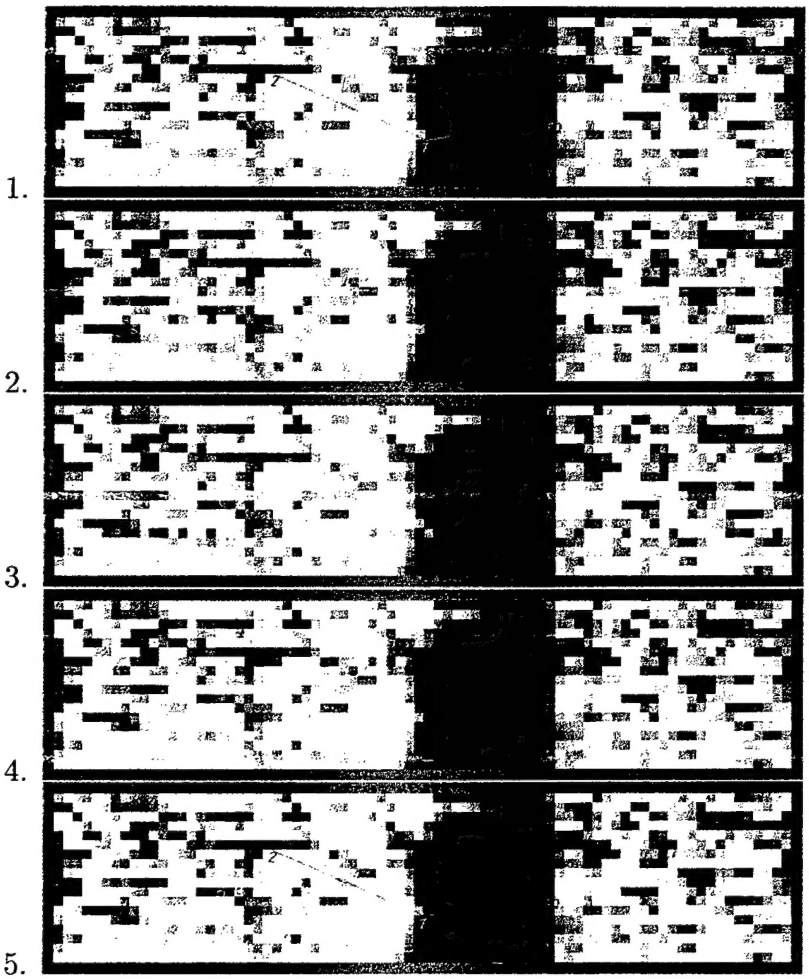

Figure 5: From top to bottom: 1. Graph on a digit '6' at node 011 in a tree. 2. Tags 103 to 107 on the same data. 3. Tags 118 to 123. 4. Tags 58 to 62. 5. Graph on same digit at node 01101 in the same tree.

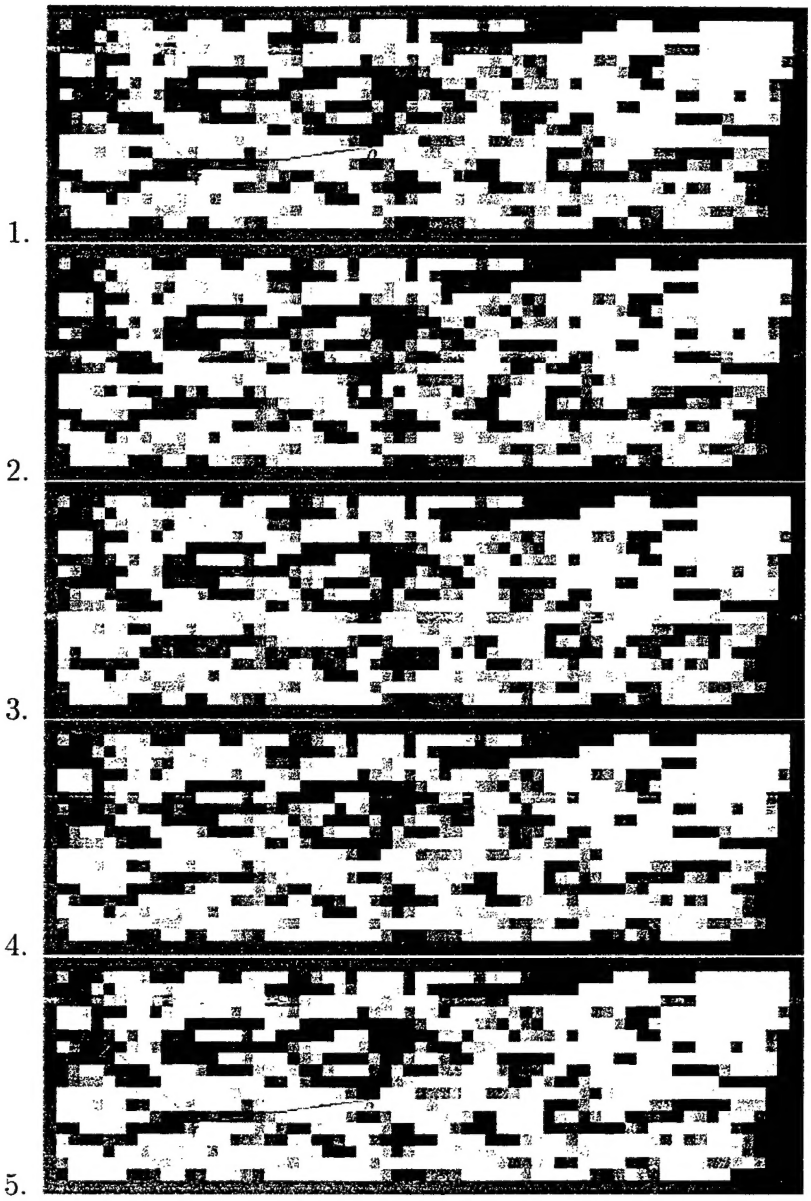

Figure 6: From top to bottom: 1. Graph on a digit '3' at node 011 in a tree. 2. Tags 103 to 107 on the same data. 3. Tags 118 to 123. 4. Tags 58 to 62. 5. Graph on same digit at node 01100 (same graph) in the same tree.

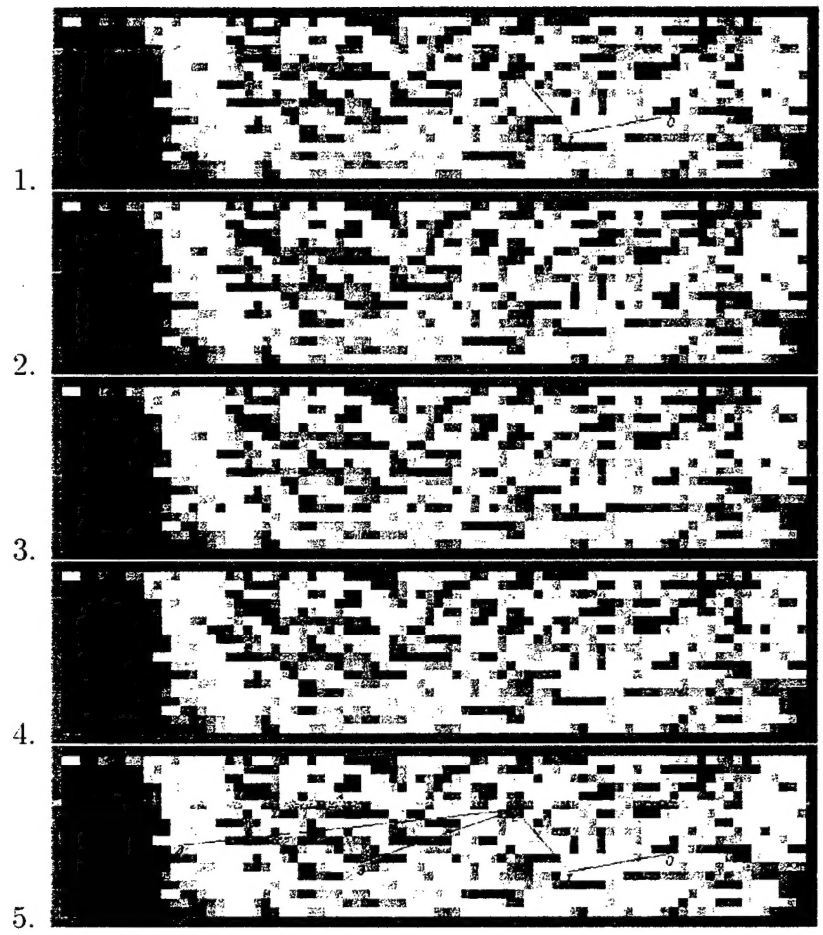

Figure 7: From top to bottom: 1. Graph on a digit '4' at node 011 in a tree. 2. Tags 103 to 107 on the same data. 3. Tags 118 to 123. 4. Tags 58 to 62. 5. Graph on same digit at node 01111 (same graph) in the same tree.